

HIT137

Group Assignment 2 (20% Mark)

“assignment2.zip” contains all the files for this assignment.

You are required to create a GitHub repository and add all your group mates to it (make sure to keep it public, not private). You should do this before you start the assignment.

All the answers and contributions should be recorded in GitHub till you submit the assignment.

Submission Guidelines:

- Include your GitHub Repository link in a text file “github_link.txt”.
- Zip all the programming files and outputs and “github_link.txt” and upload them to Learline.

Question 1

Create a program that reads the text file “raw_text.txt”, encrypts its contents using a simple encryption method, and writes the encrypted text to a new file “encrypted_text.txt”. Then create a function to decrypt the content, and a function to check the correctness of decrypted text.

Requirements

The encryption should take two user inputs (n, m), and follow these rules:

- For lowercase letters:
 - If the letter is in first half of alphabet (a-m): shift forward by $n * m$
 - If the letter is in second half (n-z): shift backward by $n + m$
- For uppercase letters:
 - If the letter is in first half (A-M): shift backward by n
 - If the letter is in second half (N-Z): shift forward by m^2
- Special characters, and numbers remain unchanged.

Question 2

Create a program that analyses temperature data collected from multiple weather stations in Australia. The data is stored in multiple CSV files under a temperatures folder, with each file representing data from one year.

You need to:

- Calculate the average temperatures for each season across all years. Save the result to file “average_temp.txt”.
- Find the station/stations have the largest temperature range. Save the result to file “largest_temp_range_station.txt”.
- Find the warmest and coolest station/stations. Save the result to file “warmest_and_coolest_station.txt”.

Question 3

Create a program that use recursive function to generates a tree pattern using Python's turtle graphics. The program should take the following parameters from the user:

Left and right branch angles

Starting branch length

Recursion depth

Branch length reduction factor

Example:

The tree with branch left at 20 degrees and right at 25 degrees, starting with a branch length of 100 pixels, and make each new branch 70% long as its parent branch, letting it branch out 5 times.

Example output:

